IOS XR Software
CRS-1 and C12000

Martin Winter

Technical Leader

Agenda

- High Level IOS XR Strategy
- IOS XR Software Architecture
- IOS XR CLI

High level IOS XR Strategy

High End Routing Portfolio

CRS-1

Next Generation Core

- 40G Routing Day 1
- Multi-Chassis Scale
- Foundation for Core Consolidation

Cisco XR 12000

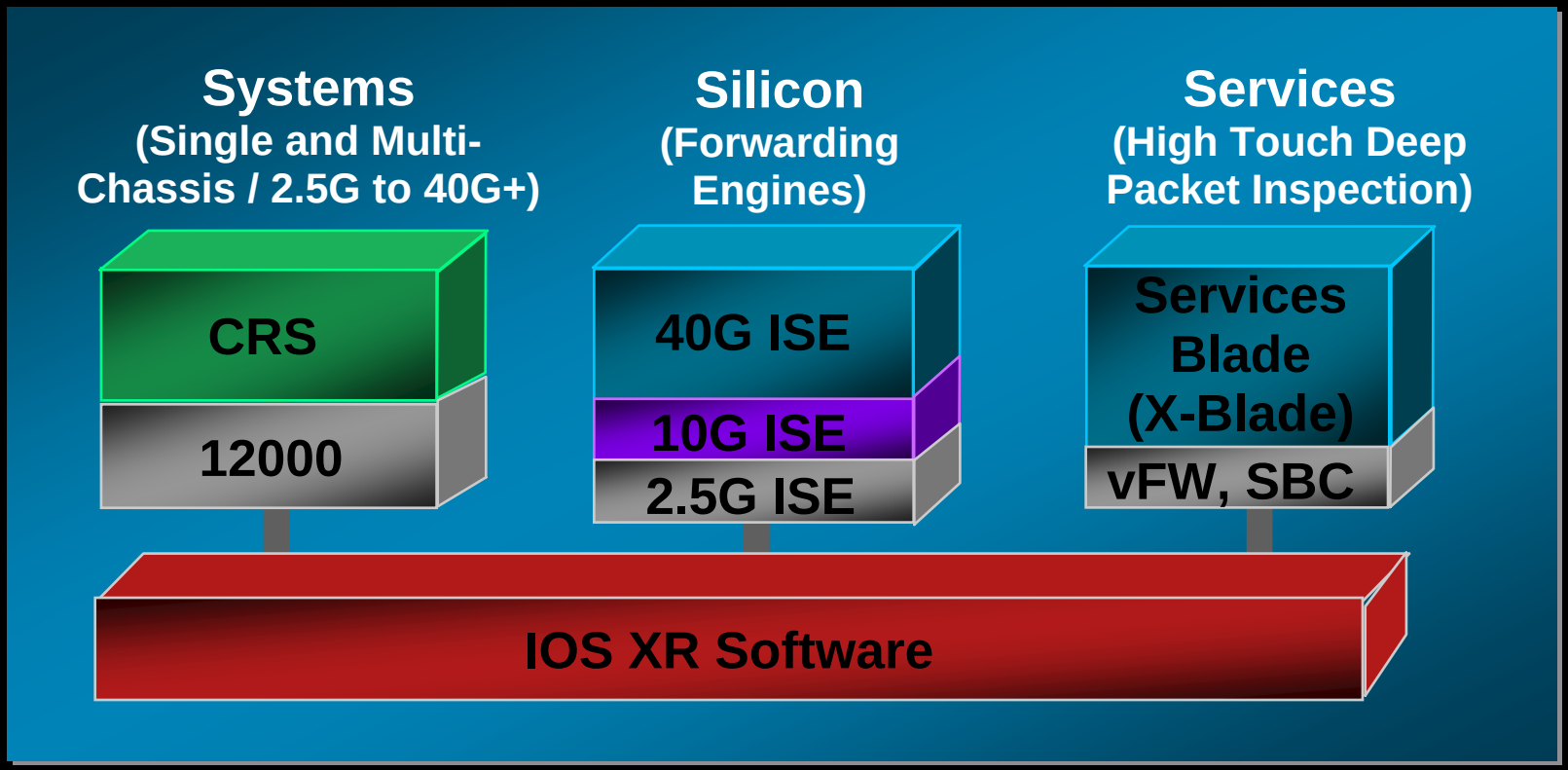
Next Generation Core & Edge

- **Builds on 12000 Series Technology**
  - **PRP, 2.5G ISE, 10G ISE**
- **Edge interface breadth/density**
- **4/6/10/16 Slot Form Factor**
- **Foundation for Multi-Service Edge consolidation**

Cisco High End Routing Strategy

IOS XR: Foundation of Cisco HER Technology Convergence

High End Routing Platforms



- **IOS XR is the 'glue', delivering HA, scale, core+edge services with common management and user interface**

IOS XR Software Architecture

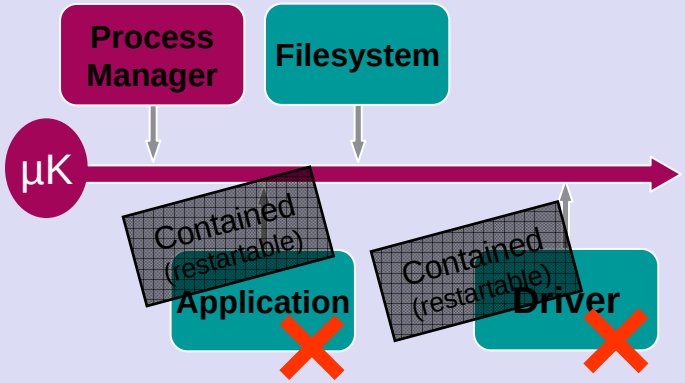
Modular IOS != IOS XR

- Modular IOS:
  - Ships today on Catalyst 6500 with Sup720 and Sup32
  - Based on the same IOS code with added Microkernel and IOS split into multiple processes.
  - Not everything as it's own process (ie all Routing as one process), optimized for performance on existing hardware
- IOS XR:
  - Ships today on CRS-1 and C12000 (PRP only)
  - Complete rewrite of the code
  - Very modular, split into multiple processes and built for multiterabit scaling and distributed operation
  - Features targeted for SP NGN router

The Microkernel, the foundation of IOS XR

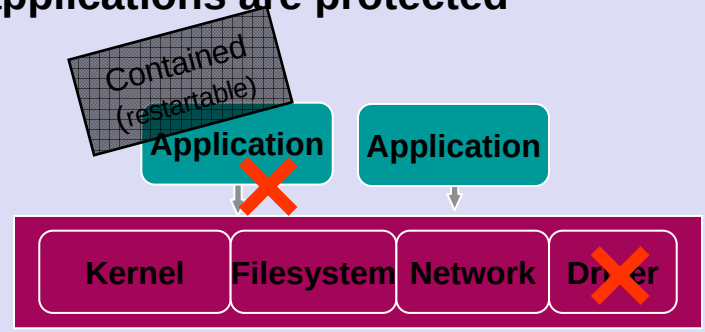
TRUE Microkernel (Mach, QNX)

- MMU with full protection for protected Applications, drivers, and protocols



Monolithic Kernel (BSD/Linux, NT)

- MMU with partial protection. Only applications are protected

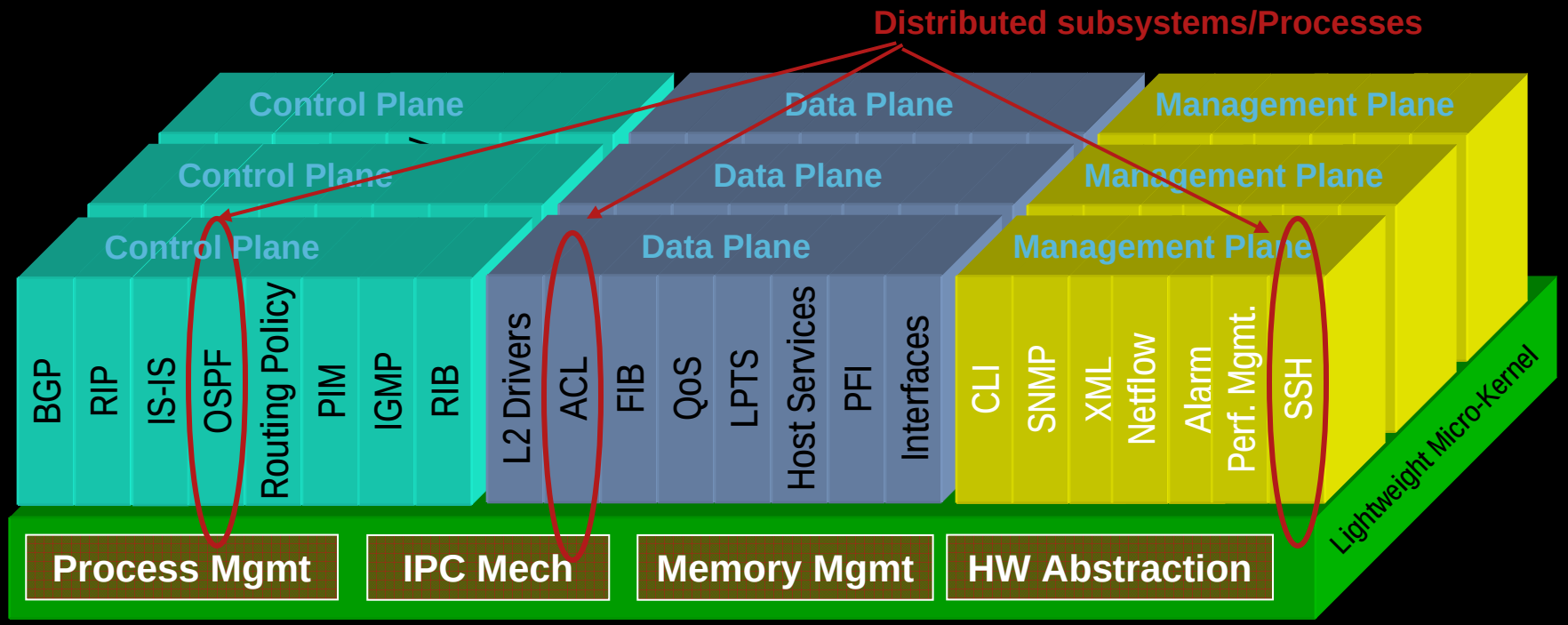


System wide corruption - Router Restart

Feature	Microkernel	Monolithic Kernel
Preemptive scheduler with support for process priority	Yes	Yes
Protected memory architecture for application processes	Yes	Yes
Protected memory architecture for system processes	Yes	NO
Fault protection for application processes	Yes	Yes
Fault protection for Host Stack	Yes	NO
Fault protection for device drivers	Yes	NO
Fault protection for file system	Yes	NO
In Service SW Upgrade for application processes	Yes	Yes
In Service SW Upgrade for Network Drivers, File System	Yes	NO

IOS XR Software Architecture

Modular, Distributed Architecture



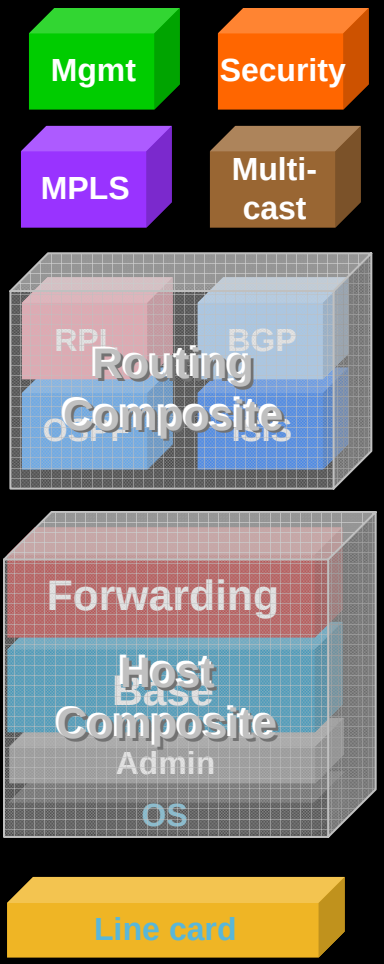
IOS XR Architecture Features

- Real Time Deterministic Scheduling
- Full Memory Protection
- Light weight Microkernel
- Restartability
- Patchability
- True Modularity
- Distributed Processes/subsystems
- Virtualization
- Checkpointing for stateful recovery

IOS XR Architecture Benefits

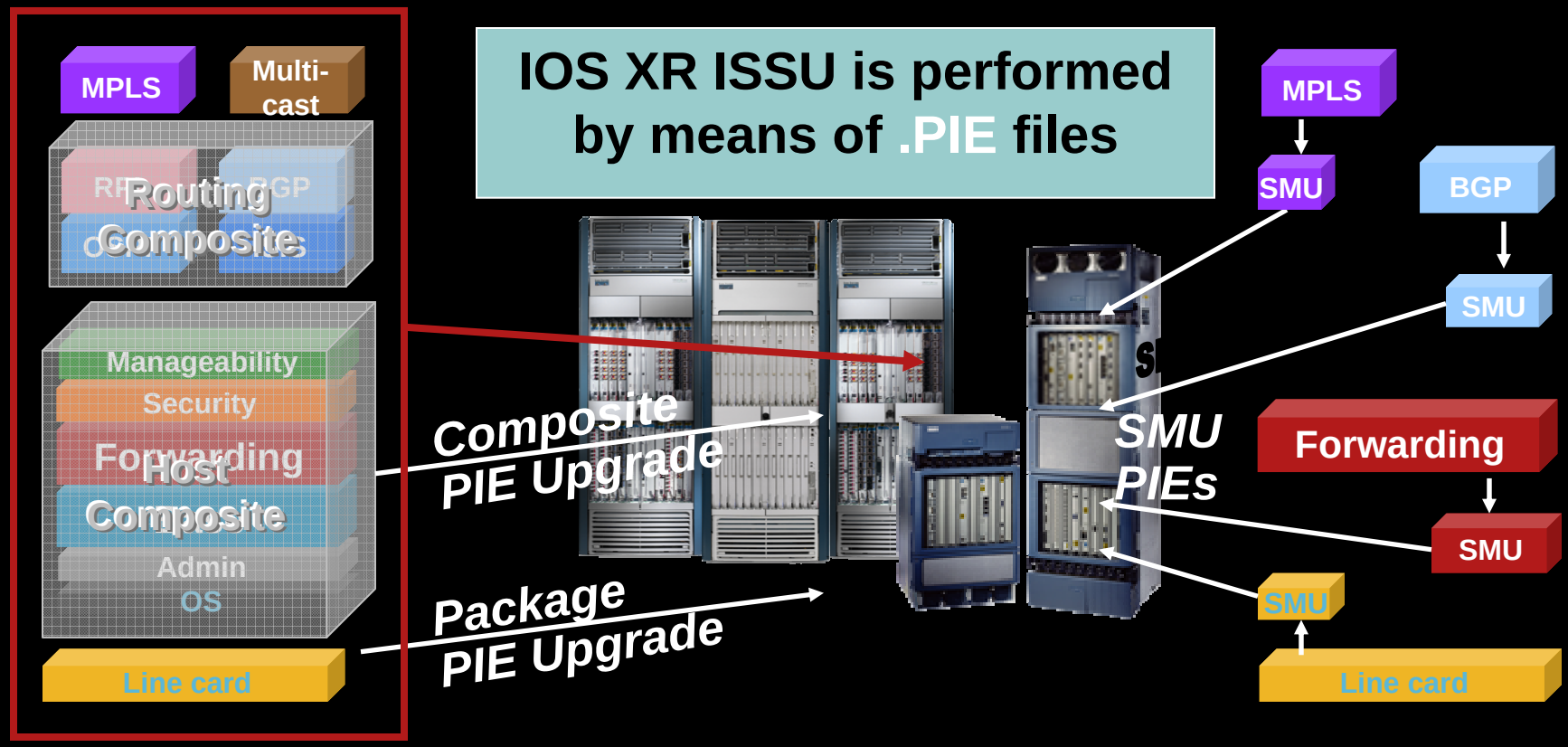
- Reliable architecture enabling highly available applications
- Distributed to enable high level of scale limited only by hardware
- Feature velocity due to modular software design

IOS XR Modular Software Packaging



- Code base files are organized into components – these are versioned and visible to the development engineer
- Packages are unique sets of components and represent *potential* units of delivery
- Packages are visible in the code base – "build" infrastructure prevents illegal dependencies between packages
- Packages can be grouped into composites for ease of delivery
- SW is packaged and can be upgraded along these Composites:
  - Host **– includes Microkernel, Infrastructure code, platform independent forwarding code, host stack**
  - Line Card **– Line card specific drivers and platform code**
  - Routing **- Support for static & dynamic unicast routing**
  - Multicast **- Support for Multicast protocols**
  - MPLS **– MPLS, GMPLS, & UCP functionality**
  - Mgmt **– XML, CWI**
  - Security **– non-exportable security features**

In Service Software Upgrades (ISSU)



- Upgrades can be on Composite, Package, or SMU boundaries
- Upgrades are performed in-service
- Upgrades can be rolled back
- Software Maintenance Updates (SMU) or patches provide pointed corrections for mission critical defects
- Line cards upgrades can be independent of Route Processor

IOS XR Carrier Class High Availability

Built for Non-Stop Operations

99.999+% Service Availability

ISSU
In Service Software Upgrade

Non-Stop Forwarding

Process Restartability with Active State Checkpointing

Protected Memory Processes
Memory faults affect only 1 process

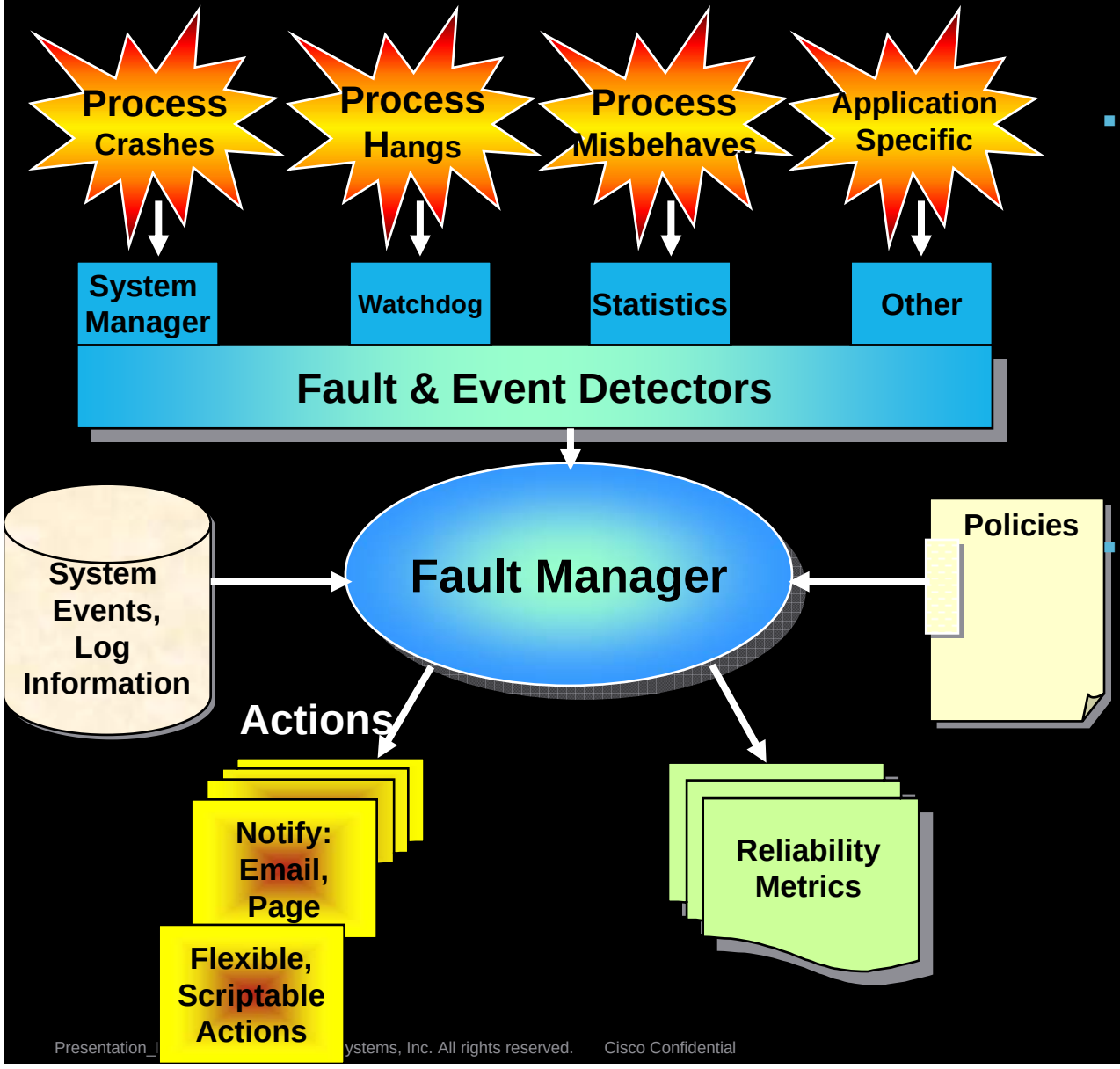
Software Design: Highly Modular, Separation of Control, Data, Management Planes, Fault Management, MicroKernel, Packaging Model

Hardware Design: Redundancy (Fabric, Power, Thermal, Route Processor, Line Card), High MTBF, Distributed Forwarding, Online Insertion Removal (OIR), Parity or Error Correcting Memory, Fault Insertion Testing

- Graceful Restart
  - Shipping: OSPF (Cisco), ISIS, BGP, LDP, RSVP-TE, Multicast
- Line Card Redundancy
  - Shipping: 1+1 SONET/SDH APS
- HA Components
  - Shipping: IOS XR, MPLS TE FRR
- Software Upgrades
  - Shipping: ISSU (Patching), SMU

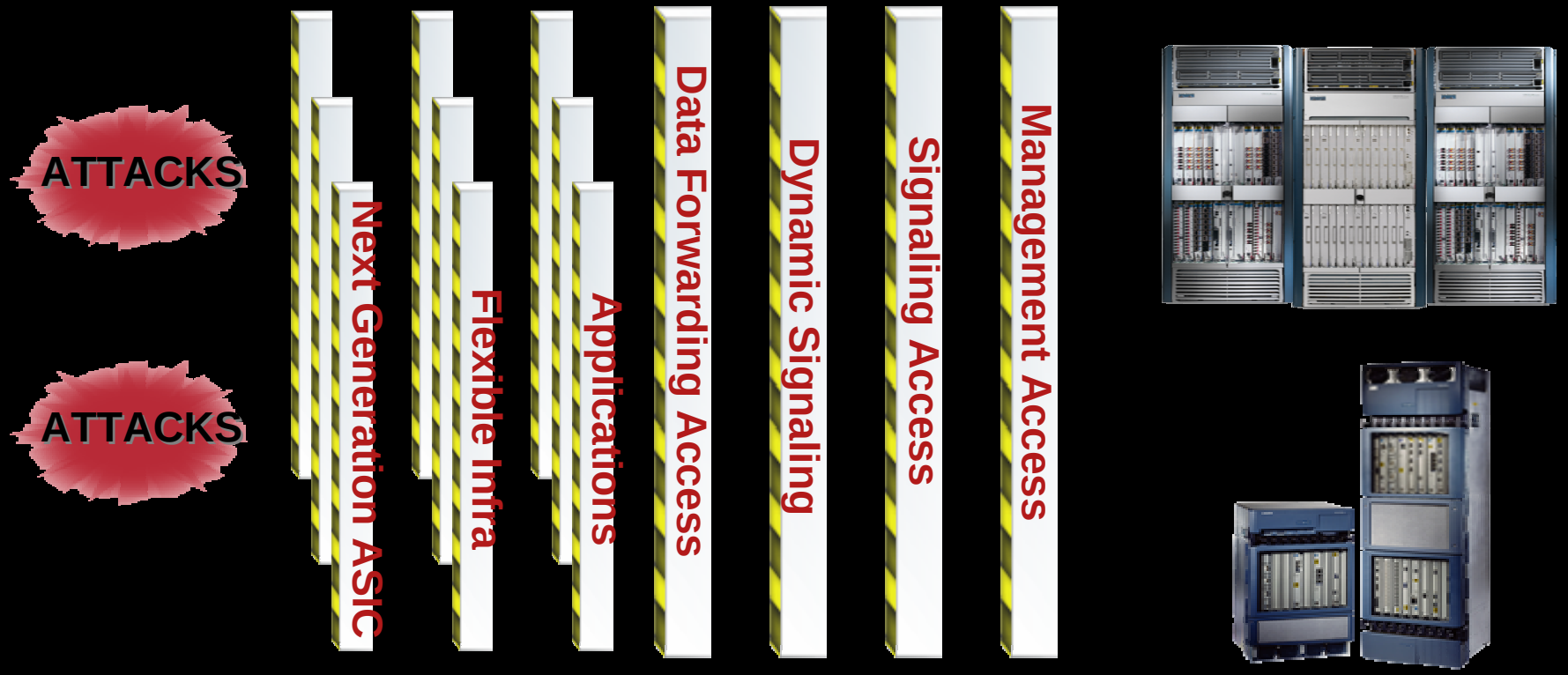
IOS XR Fault Management

Error Monitoring and Reporting



- Fault Manager checks for established policy handlers:
  - If a policy handler exists, the FM runs the policy (TCL script) that implements recovery actions.
  - If a policy handler doesn't exist, the system performs a built-in default action defined for this event type (if any).
- Example:
  - Default action for a process fault is automatic restart. It's defined in startup files by developers and can't be set by users.
  - Users can enhance the default action by writing an FM policy.

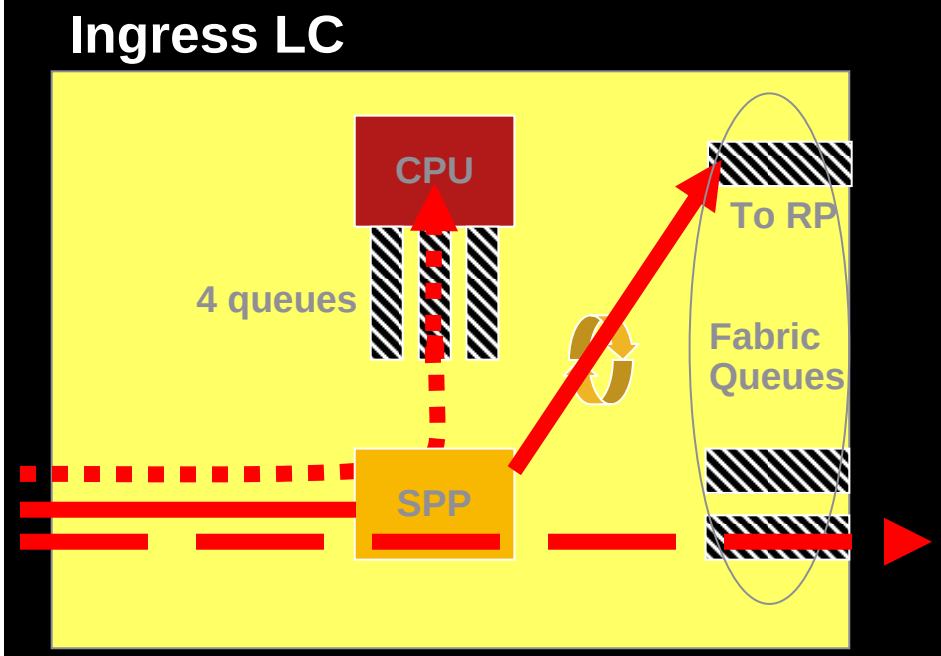
IOS XR Carrier Class Security



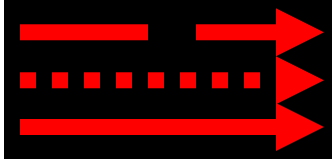
- IOS XR provides a layered approach for total system security
- IOS XR Architecture and coupled with the CRS-1 and Cisco 12000 hardware design provides the foundation for secure networking applications
- Protection is completed with IOS XR's security aware management access, signaling access, and router applications

IOS XR Carrier Class Security

Data Forwarding Access Security

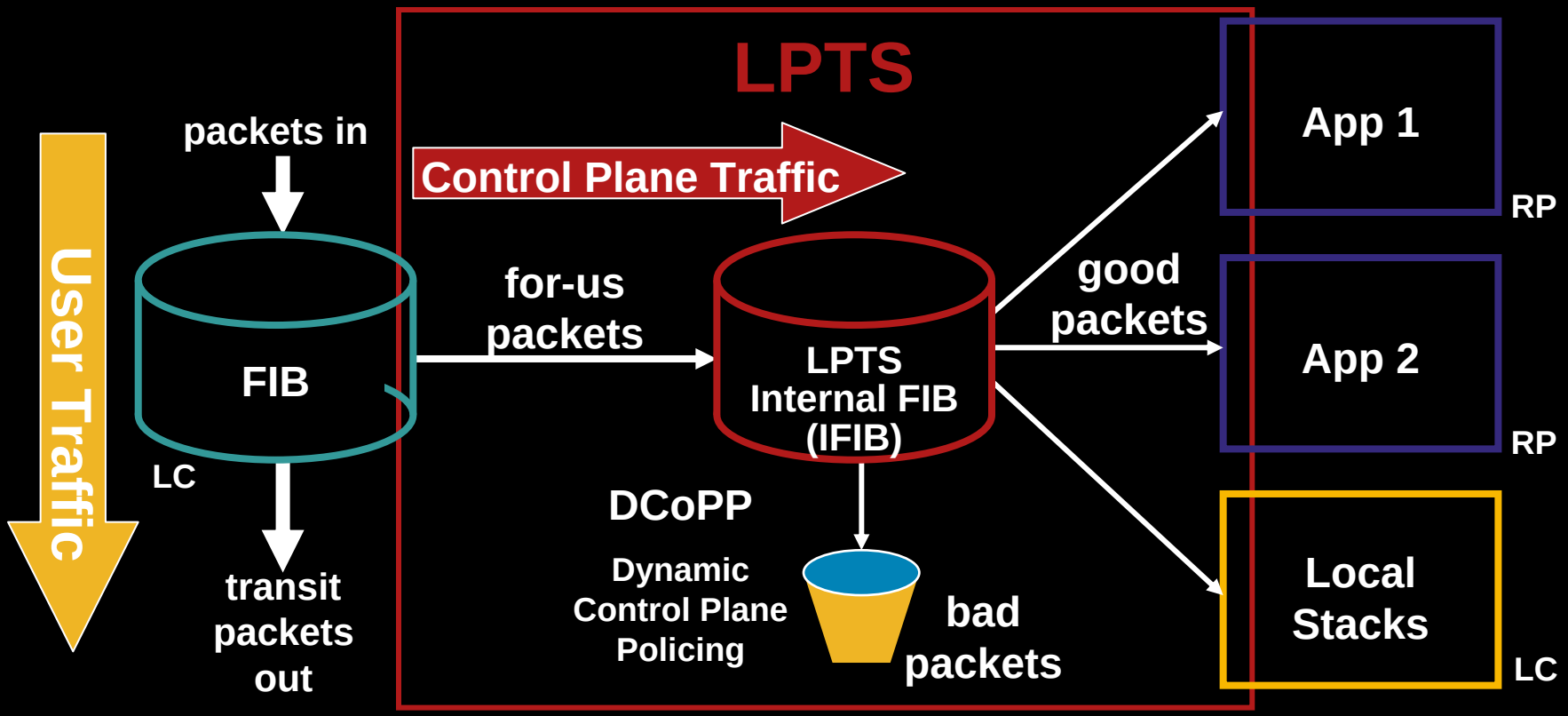


- Control plane classification, policing, and queuing provide the foundation to stopping DoS attack
- 4 queues into LC CPU
  - Low: TTL errors, Options, logging, ICMP
  - Medium: IPv4 lookup
  - High: ARP
  - Critical: Layer 2 keep alive. (PPP, HDLC)
- 3 queues into RP CPU
  - Low: other
  - Medium: BGP, PIM, LDP, SSH
  - High: OSPF, ISIS
- Priority queuing among software queues



Transit User Traffic
ICMP, IP Options logging, …
Control Plane (routing, labels)

IOS XR Control Plane
Local Packet Transport Service



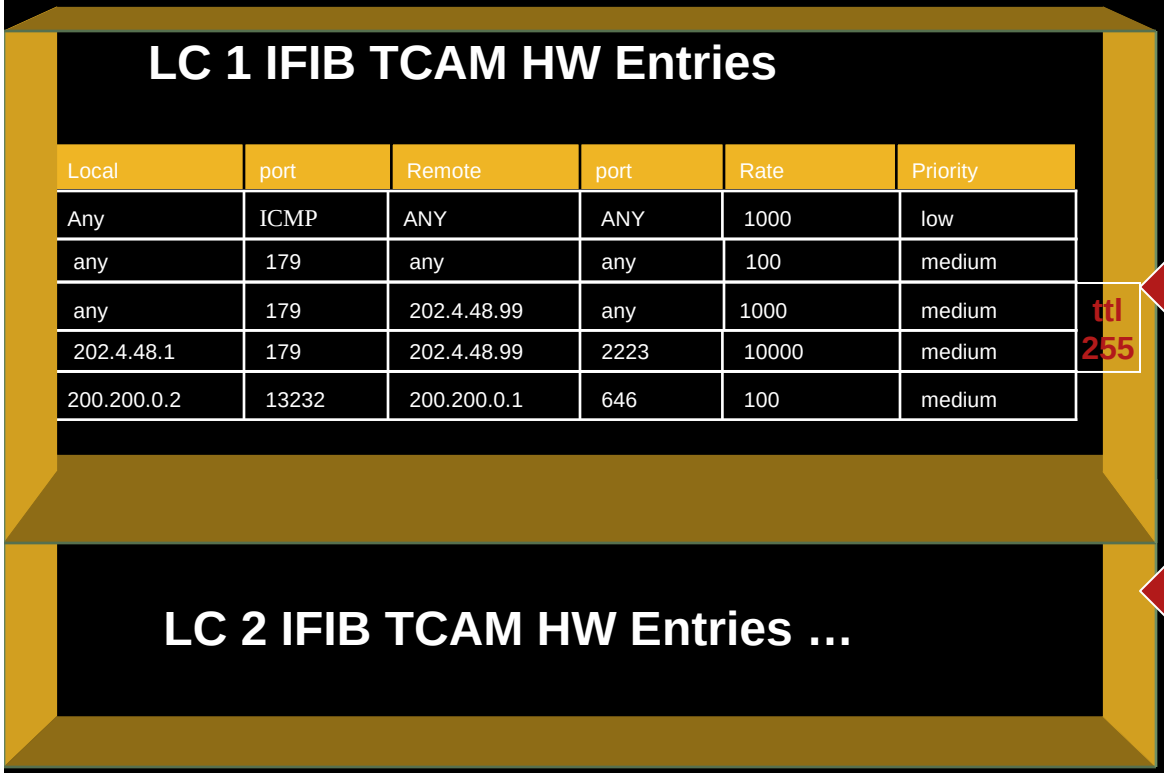
- LPTS enables applications to reside on any or all RPs, DRPs, or LCs
  - Active/Standby, Distributed Applications, Local processing
- IFIB forwarding is based on matching control plane flows
  - DCoPP is built in firewall for control plane traffic.
- LPTS is transparent and automatic

IOS XR LPTS

Dynamic Control Plane Protection

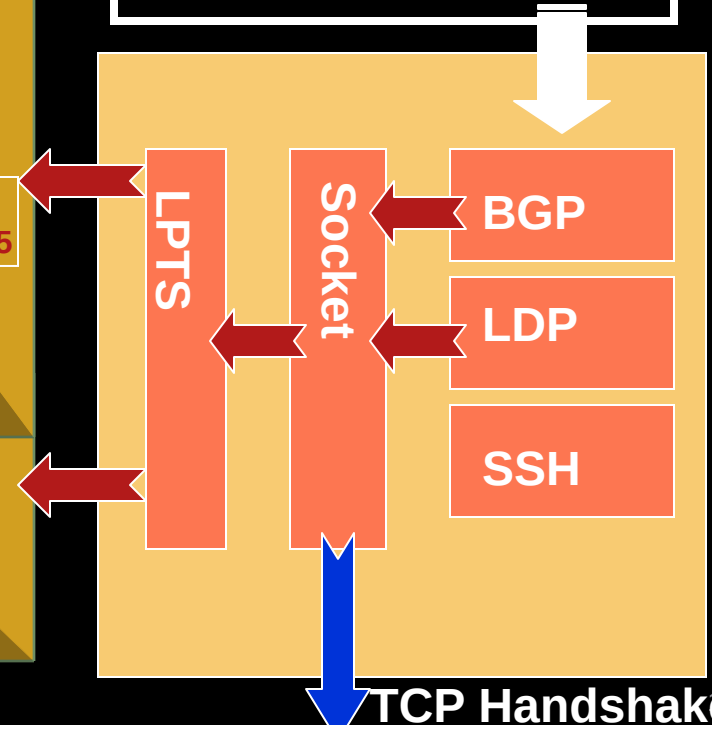
- DCoPP is an automatic, built in firewall for control plane traffic.
- DCoPP is being made user configurable

```
Router bgp
  neighbor 202.4.48.99
  ttl_security
!
mpls ldp
  …
!
```



Local	port	Remote	port	Rate	Priority
Any	ICMP	ANY	ANY	1000	low
any	179	any	any	100	medium
any	179	202.4.48.99	any	1000	medium
202.4.48.1	179	202.4.48.99	2223	10000	medium
200.200.0.2	13232	200.200.0.1	646	100	medium

ttl 255



IOS XR Carrier Class Security

Signaling Access Security

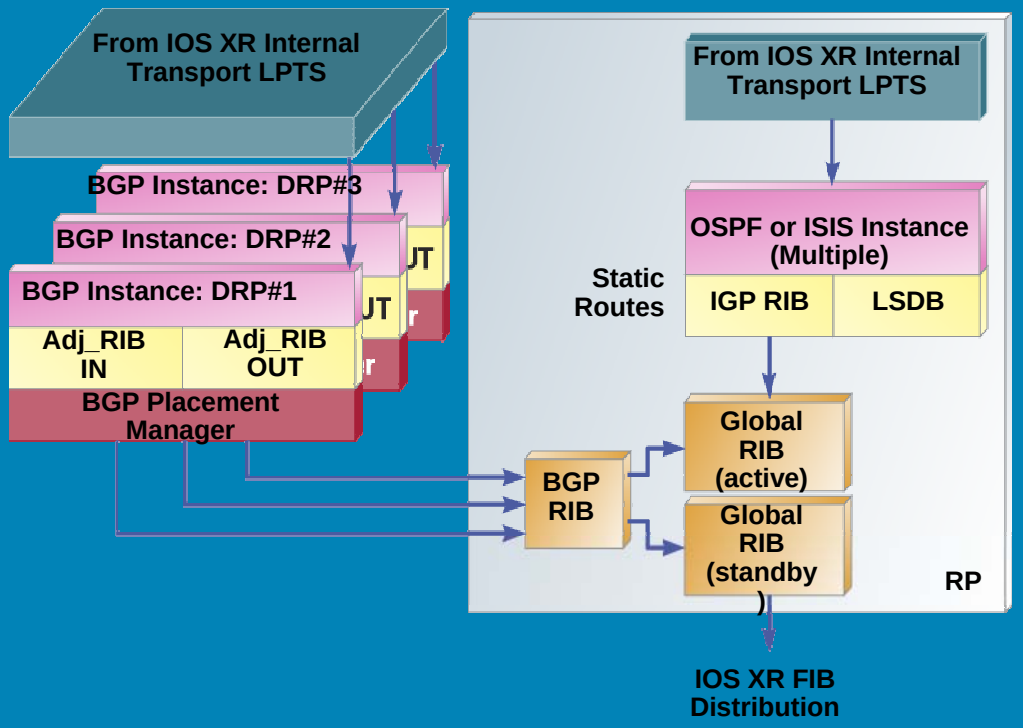
- Support MD5 authentication for routing protocols
  - BGP, ISIS, OSPF, LDP, RSVP
- Support GTSM RFC 3682 (formerly BTSH)
- Open services off by default
- Security stress testing and audits through SAT, SecureScanX, Nessus, Datapool tests.
- Developed IOS XR with Cisco Security experts from PSIRT, NSITE, Alcazar, ARF, STAT teams… to learn and share experiences.

IOS XR Carrier Class Security
Management Access Security

- Support for SSH, SSL, SCP, IPSEC, IKE
- Support for SNMPv3
- Authenticated software installation
  - Only authorized software can be installed
- Role based User Management
  - Using TACACS+ for CLI and XML interfaces
  - Administer EMS user(s)/roles/responsibilities
  - Administer NE user(s)/roles/responsibilities
- Logging and auditing
  - Maintain log of security events
  - system access, unauthorized attempts, profile changes, etc.)
  - Support audit tools to produce exception, summary and detailed reports

IOS XR Distributed Processing
Distributed Control Plane

Multi-Speaker BGP



- IOS XR supports multiple (D)RPs per system
  - Logical Routers
  - Additional processing capacity
  - Routing protocols and signaling protocols can run in one or more (D)RP
  - Dedicated Management RP
- Each (D)RP can have redundancy support with standby (D)RP

For example, Multi-Speaker BGP for high scale applications

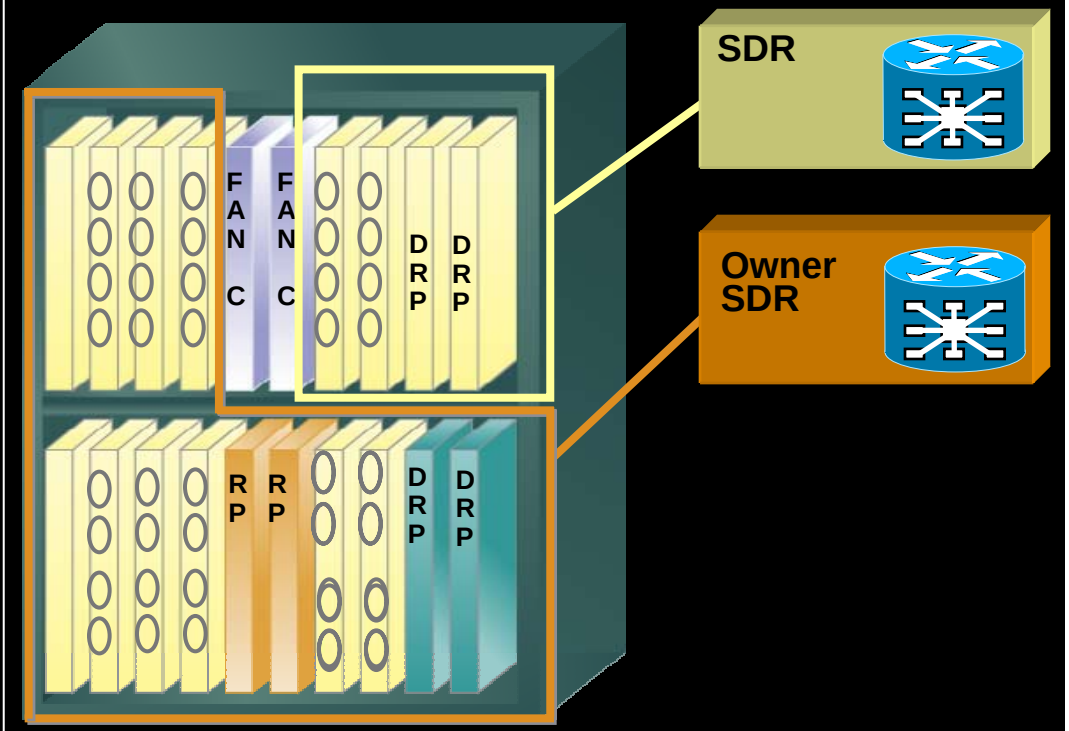
- Distributed BGP speakers to multiple RP and DRPs
- Single unified BGP RIB to external peers
- Achieve BGP peering scalability (many 1000s of peers)

IOS-XR Service Separation Architecture

Resource Partitioning/Sharing, with Admin & Fault Isolation

- Physical Separation:
  Secure Domain Routers (SDR)
  - Isolated physical routing instances with independent management, control, and data planes.
  - Defined by a subset of RPs and LCs within a common (Multi-)chassis.
  - Dedicated RPs, RIB, FIB, Memory, CPU, interfaces
  - SDRs definable across multichassis boundaries
  - SDRs share redundant cooling, power, fabric and the (multi-)chassis.

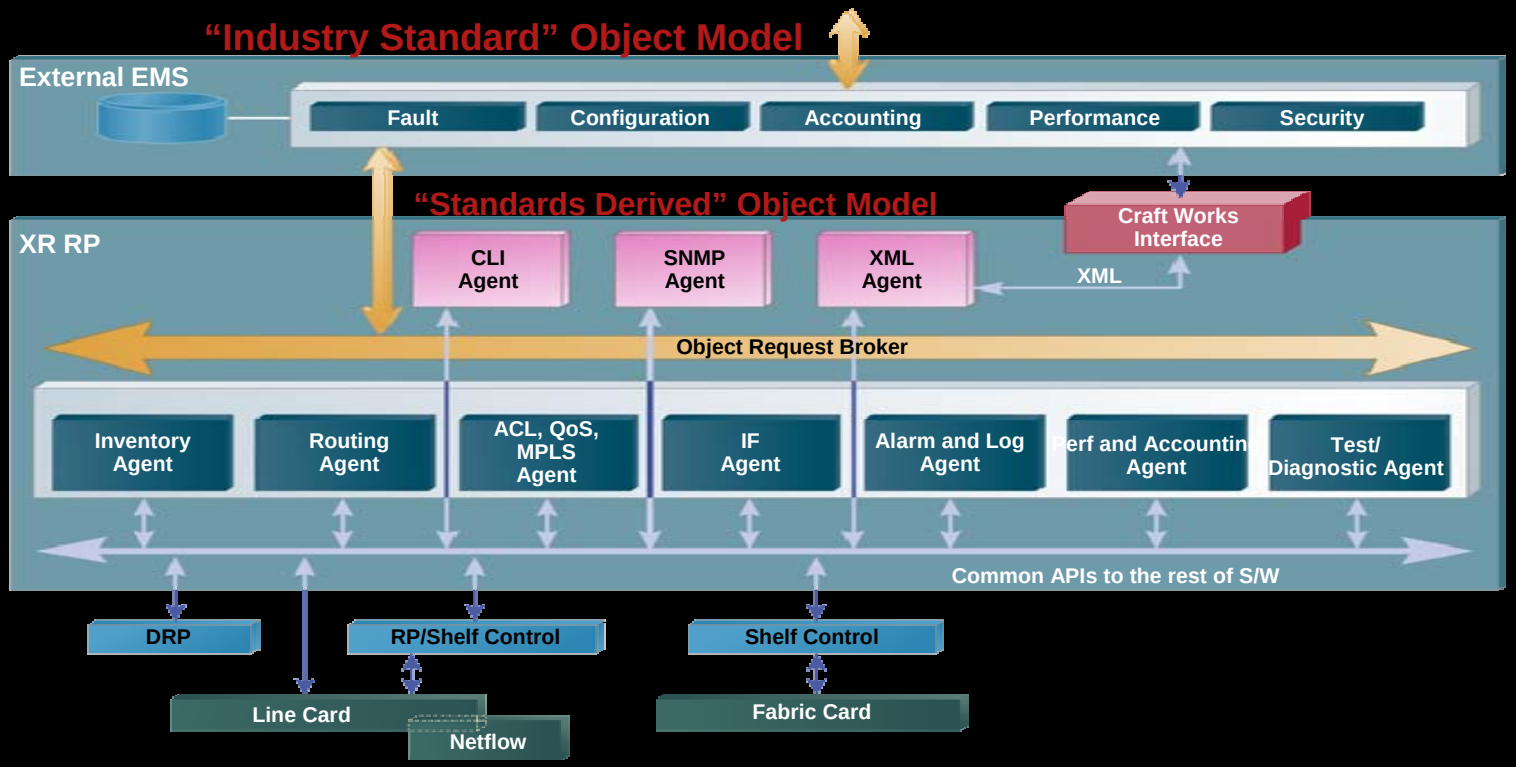
- SDR Benefits
  - Single-system simplicity, with multi-box fault and administrative isolation.
  - Additional dRPS can be added - in service - to increase control plane scale of any SDR
  - dRPs and LCs can be dynamically reassigned to meet changing service & b/w needs
  - Per SDR ISSU supported to allow new features in one SDR without impacting others
  - All Routing features supported for service flexibility. No feature caveats.



Secure Separation Architecture (SSA)

Key Router Instantiation Feature	Cisco SDR	Generic Virtual Routers
Every feature of the router is supported	YES	No
Different Software releases allowed per Instantiation	YES	No
Per Instantiation ISSU	YES	No
Per Instantiation software packaging	YES	No
Full hardware/software isolation between Instantiations	YES	No
Distributed Processing Support – for additional scale and processing capability	YES	No
Fully Separated Control Planes - anomalies in one instantiation do NOT affect other instantiations	YES	No, shared
Fully Separated management plane – complete administrative separation	YES	No, Centralized
Mis-configurations are isolated per instantiation	YES	No, Shared
Dynamically reassignable resources	YES	YES

IOS XR Manageability



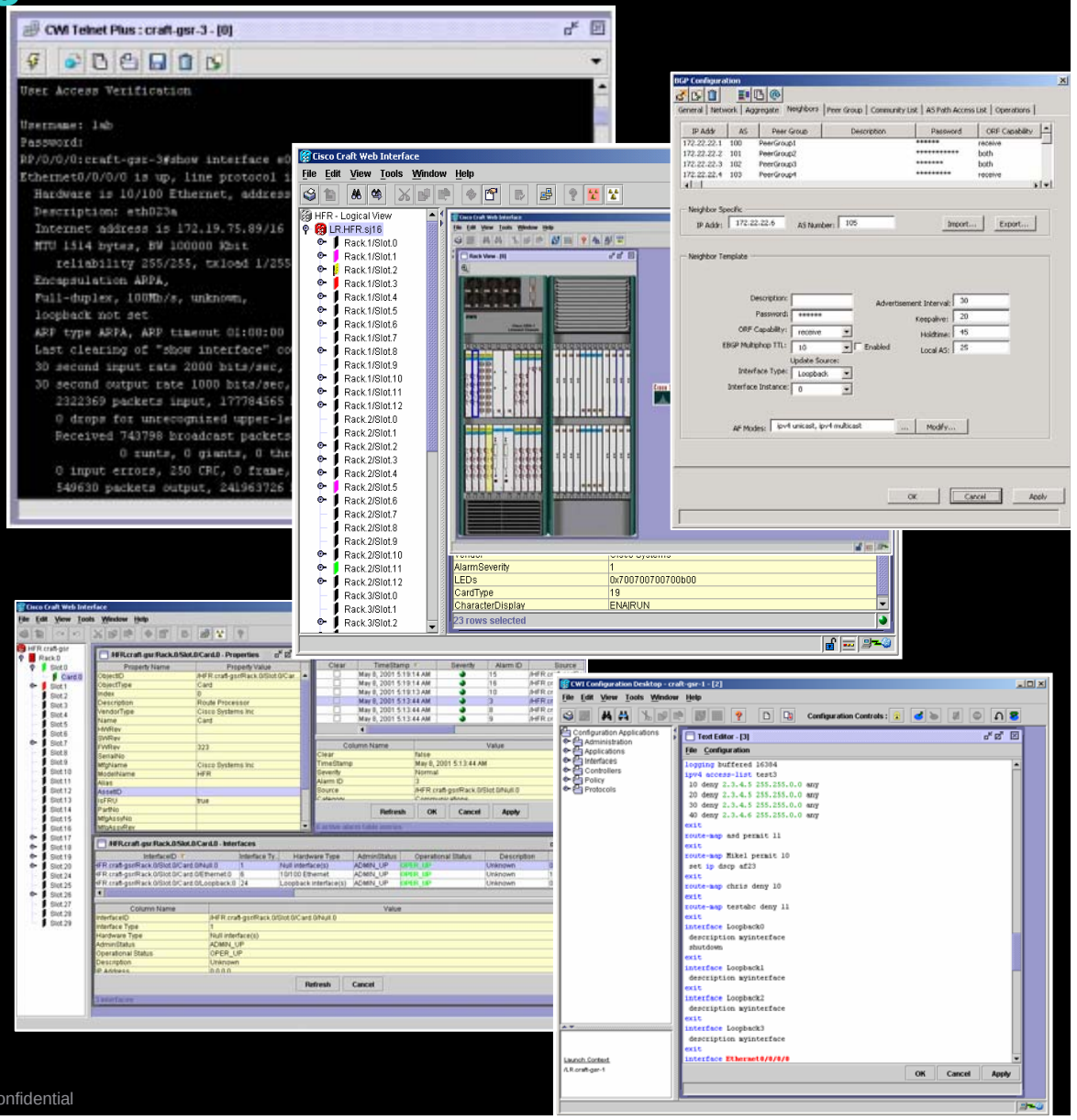
- Consistent data model independent of access schemes: CLI, SNMP or XML
  - Embedded Agents for command and control
  - Programmatic Interfaces – XML/CORBA; SNMP
  - Traditional Command Line Interface – CLI
- Software Development Kit (SDK) provides smooth backend OSS/EMS integration

- Element Management System (EMS)
  - Fault, Configuration, Accounting, Performance & Security
  - Data Collection, Storage, and Historical Reporting "Standardized" mediation to external systems

IOS XR's Craft Works Interface (CWI)

Industry Leading User Interface

- Java application launched from web browser
- Interacts with the Router's XML
  - Metadata for fast feature development
- Graphical Configuration Desktop
  - Provides traditional CLI through CWI Telnet+
  - Config Validation with 2 stage configuration
  - Embedded Configuration Text Editor
  - Value-added SSH/Telnet
  - Inventory and Rack View
  - Integrated Alarm Views
- Increased Operator Productivity

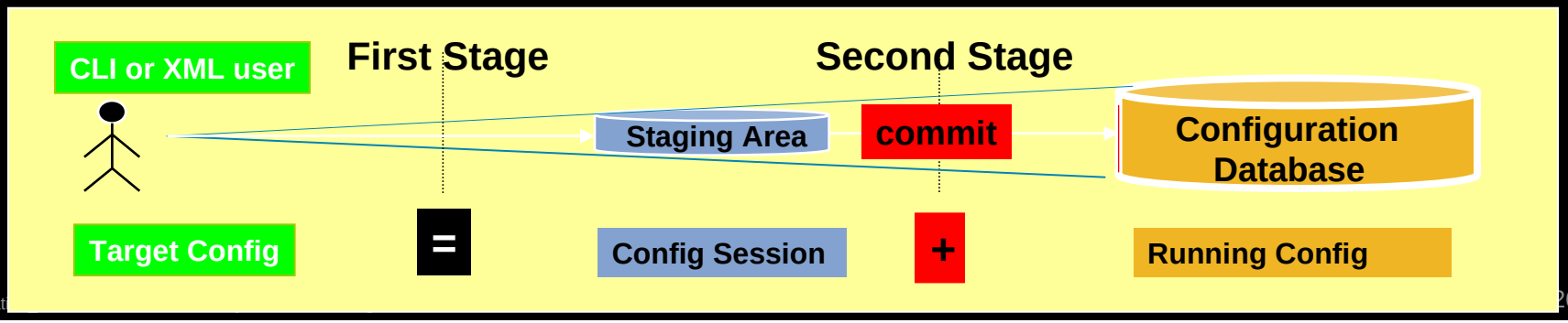


IOS XR CLI

IOS XR's CLI Configuration Model

Two Stage Configuration

- Configuration first enters a staging area (first stage)
  - Users and their commands are authorized in staging area to limit operator to their administrative role
  - Offline configuration and syntax checks eliminates operator errors during configuration
  - Active Configuration can not be modified directly
- Configuration must be explicitly promoted to become part of the active configuration (second stage).
  - Configuration audit log kept to track when, who, and why changes were made
  - Rollback available to easily to revert to any of the last 20 configurations
  - Change Notification generated to syslog to track configuration changes



IOS XR CLI: Config Commits

```
RP/0/0/CPU0:ios#show run int gi0/2/0/0
% No such configuration item(s)

RP/0/0/CPU0:iosxr1#conf t
RP/0/0/CPU0:iosxr1(config)#interface gig0/2/0/0
RP/0/0/CPU0:iosxr1(config-if)#ipv4 address 100.12.1.1/24
RP/0/0/CPU0:iosxr1(config-if)#commit
RP/0/0/CPU0:Apr 24 00:49:28.119 : config[65691]: %MGBL-
CONFIG-6-DB_COMMIT : Configuration committed by user 'root'.
Use 'show configuration commit changes 1000000036' to view
the changes.
RP/0/0/CPU0:iosxr1(config-if)#end
RP/0/0/CPU0:Apr 24 00:49:30.701 : config[65691]: %MGBL-SYS-
5-CONFIG_I : Configured from console by root
RP/0/0/CPU0:iosxr1#
RP/0/0/CPU0:iosxr1#show run int gigabitEthernet 0/2/0/0
interface GigabitEthernet0/2/0/0
 ipv4 address 100.12.1.1 255.255.255.0
```

ISIS/OSPF CLI Differences

IOS XR ISIS Configuration:

```
router isis IOS XR
 net 47.1111.1111.0001.0000.0c00.0006.00
 nsf ietf

 interface POS0/4/0/0
  address-family ipv4 unicast
  !
 !
!
```

IOS ISIS Configuration:

```
router isis IOS
 net 47.1111.1111.0001.0000.0c00.0006.00
log-adjacency-changes
 nsf ietf
!
interface POS1/0/0
 ip address 201.1.1.2 255.255.255.0
 ip router isis IOS
```

IOS XR OSPF Configuration

```
router ospf 99
 router-id 1.1.1.1
 area 0
  interface GigabitEthernet0/2/0/0
     !
```

IOS OSPF Configuration

```
router ospf 99
 router-id 1.1.1.1
 log-adjacency-changes
 network 201.0.0.0 0.0.0.255 area 0
!
interface POS1/0/0
 ip address 201.1.1.2 255.255.255.0
```

Comparison of Cisco IOS Static Route and Cisco IOS XR Static Route

Static Route IOS

```
IOS#sh run | beg ip route 192.1.1.0

ip route 192.1.1.0 255.255.255.0 g4/0
ip route 223.255.254.0 255.255.255.0 10.13.0.1
```

Static Route IOS XR

```
RP/0/1/CPU0:IOS XR#sh run router static

router static
 address-family ipv4 unicast
  43.43.44.0/24 Serial0/5/3/3/0:2
  43.43.44.44/32 Serial0/5/3/3/0:0
  223.255.254.254/32 MgmtEth0/1/CPU0/0
 !
 address-family ipv6 unicast
  5301::1111/128 Serial0/5/3/3/0:0
 !
!
```

Comparison of Cisco IOS BGP and Cisco IOS XR BGP

IOS BGP Configuration

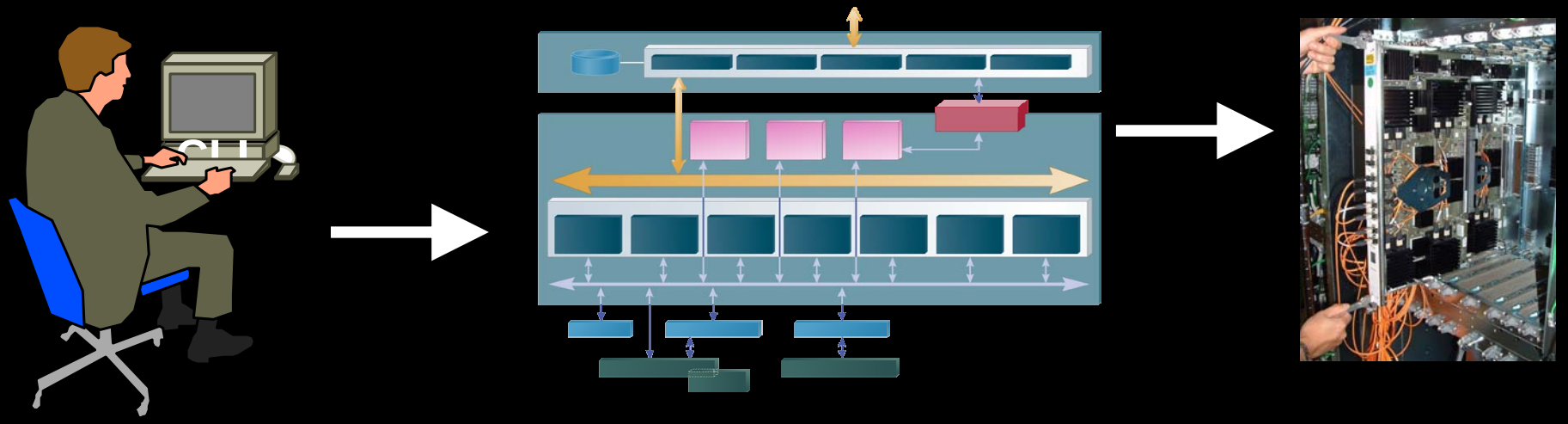
```
router bgp 1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source Loopback0
 maximum-paths 8
 !
 address-family ipv4
 neighbor 1.1.1.1 activate
 maximum-paths 8
 no auto-summary
 no synchronization
 exit-address-family
 !
```

IOS XR BGP Configuration

```
RP/0/1/CPU0:IOS XR#sh run router bgp
router bgp 300
 bgp router-id 2.2.2.2
 address-family ipv4 unicast
 !
 neighbor 192.1.1.2
  remote-as 400
  address-family ipv4 unicast
   route-policy policy in
   maximum-prefix 200000 75 warning-only
   route-policy policy out
  !
 !
```

IOS XR's CLI Configuration

Hardware Pre-configuration



- ***preconfigure interface POS0/0***
- Reduces down time
- Improves operational tasks
- Simplifies maintenance

IOS XR's CLI Configuration

Routing Policy Language (RPL)

- Routing Policy Language (RPL)
  - A "C"-like provisioning mechanism for route policy
  - Replaces IOS's route-map configuration
- RPL is used for
  - Adding, deleting, modifying routes or attributes within the bounds of the protocol specification
  - Influencing routing decision for controlling redistribution/advertisement to another peer/neighbor
- Benefits of RPL
  - Easier to manage, maintain and troubleshoot routing policies (readability & Micro*Emacs Editor*)
    - Establish sets (groups) based on AS path, Community, Extended Community and prefix
    - Supports Conditional Statements, Boolean Operators, Hierarchal insertion
  - Reduces the redundancy in route map to support large scale routing configurations.

RPL Examples

Nested conditional statements

```
if community matches(12:34, 56:78) then
    if med eq 8 then
        drop
    endif
    set local-preference 100
endif
```

Hierarchical RPL

```
route-policy rp_one
    set weight 100
end-policy

route-policy rp_two
    set med 200
end-policy

route-policy rp_three
    apply rp_two
    set community (2:666) additive
end-policy

route-policy four
    apply rp_one
    apply rp_three
    pass
end-policy
```

Boolean combinations:

med eq 10 and not destination in (10.1.3.0/24) or community is (56:78)
med eq 10 and (not destination in (10.1.3.0/24)) or community is (56:78)

show process command output

```
RP/0/RP0/CPU0:CRS#show process ospf
                  Job Id: 262
                     PID: 209102
          Executable path: /disk0/hfr-rout-3.3.1/bin/ospf
               Instance #: 1
               Version ID: 00.00.0000
                  Respawn: ON
            Respawn count: 1
  Max. spawns per minute: 12
             Last started: Thu Jul 20 15:39:20 2006
            Process state: Run
            Package state: Normal
        Started on config: cfg/gl/ipv4-ospf/proc/1/ord_z/config
                     core: TEXT SHAREDMEM MAINMEM
                Max. core: 0
                Placement: ON
             startup_path: /pkg/startup/ospf.startup
                    Ready: 13.338s
                Available: 17.353s
         Process cpu time: 2.702 user, 0.188 kernel, 2.890 total
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
262    1      60K  10 Receive       0:00:02:0672 ospf
262    2      60K  10 Receive       0:00:00:0012 ospf
262    3      60K  10 Receive       0:00:00:0012 ospf
```

JID and PID

Automatic restart?

Thread IDs

Process Restart Example

```
RP/0/RP0/CPU0:CRS#show proc qnet
                  Job Id: 74
                     PID: 32795
          Executable path: /hfr-os-3.3.1/sbin/qnet
               Instance #: 1
                    Args: transport=enet,conn_est_retries=3
               Version ID: 00.00.0000
                  Respawn: ON
            Respawn count: 1
…
RP/0/RP0/CPU0:CRS#process restart 74
RP/0/RP0/CPU0:CRS#process restart 74
RP/0/RP0/CPU0:CRS#show proc 74
                  Job Id: 74
                     PID: 7061531
          Executable path: /hfr-os-3.3.1/sbin/qnet
               Instance #: 1
                    Args: transport=enet,conn_est_retries=3
               Version ID: 00.00.0000
                  Respawn: ON
            Respawn count: 3
  Max. spawns per minute: 12
             Last started: Thu Aug 31 07:13:37 2006
            Process state: Run (last exit due to SIGTERM)
```

Initial respawn count shows process hasn't restarted

Restart a few times

PID changes, JID stays same

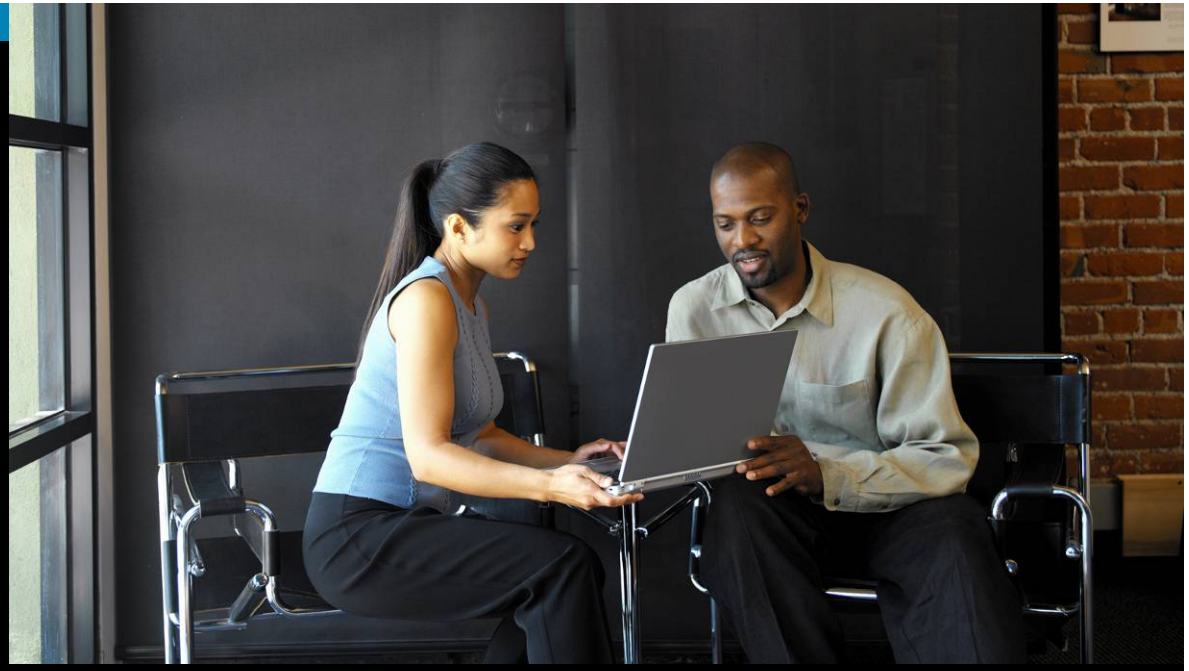
Respawn count increases

Reason for restart

Summary of IOS XR Benefits

- Modular Packaged Software Platform
  - Add/modify specific “feature packs” leading to focused certification of new features only and faster service delivery
  - Software Maintenance Upgrades (Patching) for decrease turn around time for critical software issues
- High Availability – Continuous System Availability
  - Built on a highly modularized protected memory foundation with HA features to minimize unplanned outages
  - In service Patching Eliminates DPM’s associated with Scheduled Upgrades
- Simplified OSS integration with XML/RPL and CWI
  - Leverages existing knowledge of CLI
  - Flexible Programmatic XML interface for EMS/OSS integration
  - Config staging/logging and rollback minimizes downtime due to errors
  - CWI industry leading user interface increases operator productivity
  - Role or task based authentication for increased router security
- Fully Distributed Software
  - Massive Scale increase resulting in reduced capital expense
  - Administrative and Service Separation with Logical Router (LR) and Virtual Router (VR)

Q and A



- Any questions ?